

REMARKS

Claims 1-30 are pending in the present application. By this Response, claims 1, 8, 11, 14, 18, 21 and 28 are amended. Claims 1, 11 and 21 are amended to recite subject matter similar to the canceled subject matter of claims 8, 18 and 28. Support for these amendments may be found at least on page 14, lines 5-12 of the specification. Claims 8, 18 and 28 have been amended to delete subject matter, which has been incorporated into claims 1, 11 and 21. Claim 14 is amended for clarification to recite instructions for executing the first instructions a predetermined number of times before executing the second instructions. Claim 18 is further amended to recite that the hand-off lock includes a per-node word which contains a state of the hand-off lock on each node of a multiprocessor system. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

I. 35 U.S.C. § 112, Second Paragraph

The Office Action rejects claims 14, 18, and 19 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. Claim 14 is amended to further define the instructions of the computer program product in the computer readable medium. Claim 18 is amended to further define the hand-off lock of the computer program product in the computer readable medium. Claim 19 is dependent on claim 18, which is amended to further limit the computer program product in the computer readable medium. Therefore, Applicants respectfully request the withdrawal of the rejection of claims 14, 18, and 19 under 35 U.S.C. § 112, second paragraph.

II. 35 U.S.C. § 101

The Office Action rejects claims 14, 18, and 19 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Claim 14 is amended to further define the instructions of the computer program product in the computer readable medium, which is

statutory subject matter. Claim 18 is amended to further define the hand-off lock of the computer program product in the computer readable medium, which is statutory subject matter. Claim 19 is dependent on claim 18, which is amended to further limit the computer program product in the computer readable medium, which is statutory subject matter. Therefore, Applicants respectfully request the withdrawal of the rejection of claims 14, 18, and 19 under 35 U.S.C. § 101.

III. 35 U.S.C. § 102, Alleged Anticipation, Claims 1-30

The Office Action rejects claims 1-30 under 35 U.S.C. § 102(e) as being anticipated by McKenney et al. (U.S Patent No. 6,480,918 B1). This rejection is respectfully traversed.

As to claims 1, 11, and 21, the Office Action states:

McKenney discloses the method and medium comprising means and apparatus whereof attempting to obtain a lock on a system resource (e.g., col.7, lines 31-33), associating a hand-off lock with the lock on the system resource if the attempt to obtain the lock is unsuccessful and obtaining the hand-off lock (e.g., col.7, lines 36-38).

Office Action dated October 2, 2003, pages 2-4.

Claim 1, which is representative of the other rejected independent claims 11 and 21 with regard to similarly recited subject matter, reads as follows:

1. A method of locking a system resource in a multiprocessor system, comprising:
 - attempting to obtain a lock on the system resource;
 - associating a hand-off lock with the lock on the system resource if the attempt to obtain the lock is unsuccessful, wherein the hand-off lock includes a per-processor spin field for each processor of the multiprocessor system; and
 - obtaining the hand-off lock on the system resource if the attempt to obtain the lock on the system resource is unsuccessful, wherein obtaining the hand-off lock includes spinning on a memory location identified by the per-processor spin field for an associated processor. (emphasis added)

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. In re Bond, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566,

1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. In re Lowry, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that McKenney does not identically show each and every feature of the claims arranged as they are in the claims. Specifically, McKenney does not teach associating a hand-off lock with the lock on the system resource if the attempt to obtain the lock is unsuccessful, wherein the hand-off lock includes a per-processor spin field for each processor of the multiprocessor system. Nor does McKenney teach obtaining the hand-off lock on the system resource if the attempt to obtain the lock on the system resource is unsuccessful, wherein obtaining the hand-off lock includes spinning on a memory location identified by the per-processor spin field for an associated processor.

McKenney is directed to a method of granting a lock to requesting processors that tends to keep the lock at a particular node but maintains fairness among the nodes. When a first processor releases the lock of McKenney, the lock is kept at the node if there is an outstanding lock request by a second processor at the same node, even if other processors at other nodes requested the lock before the second processor. McKenney also allows for fairness controls that prevent starvation of the other nodes by limiting how the lock is kept at the node according to some criterion.

Thus, in the system of McKenney, a locking scheme is based on a queue of requested locks and preemptive position in the queue is given to a processing unit that is spinning on the lock in the same node. Nowhere in McKenney is it taught to associate a hand-off lock with the lock on the system resource if the attempt to obtain the lock is unsuccessful, where the hand-off lock includes a per-processor spin field for each processor of the multiprocessor system. Additionally, nowhere in McKenney is it taught to obtain a hand-off lock on the system resource if the attempt to obtain the lock on the system resource is unsuccessful, wherein obtaining the hand-off lock includes spinning on a memory location identified by the per-processor spin field for an associated

processor. The Office Action alleges that these features are taught at column 7, lines 36-38, which read as follows:

If not, the lock is granted to a processing unit at another node (step 110). If so, the lock is preemptively granted to a processing unit at the same node, even if other processing units at other nodes requested the lock earlier (step 112).

In this section, McKenney is describing the kernel-level lingering lock scheme or kernel quad aware locking scheme (kqalock). There is nothing in this section that teaches associating and obtaining a hand-off lock if the attempt to obtain the lock is unsuccessful.

An example of one possible hand-off lock according to one exemplary embodiment of the present invention is a locking method where one processor hands off ownership to the next processor requesting a lock if the attempt to obtain the lock is unsuccessful. In this example, the hand-off framework defines data structures and algorithms that provide a different memory location for each processor to spin on that is requesting a given hand-off. To acquire a hand-off, a requesting processor spins on a memory location reserved for that processor until it is handed the lock by the owner. Then that processor holds the lock until it grants the lock to the next processor (see specification, pages 5-6).

To the contrary, the cited sections of McKenney actually refer to a locking method that is a kernel-level lingering lock scheme or a kernel quad aware locking scheme (kqalock). This locking scheme maintains a specialized data structure arrangement where each lock has a queue-header data structure with both a head and tail pointer to queue elements, and each queue element is associated with a particular node. A processing unit acquires a position in the queue by spinning on one of the node locks. A processing unit can acquire a preemptive position in the queue by spinning on a node queue element placed in the queue by another processing unit from the same node. For example, a processing unit could jump to the head of the queue if a processing unit from the same node is currently holding the lock. After the last active spinner for a queue element releases the lock, the second element in the queue is move to the head. Accordingly, the lock is then granted to a processing unit at the node for the new head queue element. To avoid starvation, the method adds a new queue element for a node to

the end of the queue after a queue element is used more than a predetermined number of times, leaving the old node in place.

The node lock of McKenney, which the processing unit spins on to acquire a place in the queue, is a lock on a group (node) of processors in a multiprocessor system. Thus, the processing unit of McKenney actually spins on the lock for the processors that it wants to use and is allotted a place in a queue. Additionally, a node queue element is placed in the queue corresponding to the node of the currently locked processing unit. If a lock is requested by a second processing unit and the queue contains a node queue element corresponding to the second processing unit's node, the lock request is preemptively queued at the node queue element and the second processing unit spins on the queue element. The queue element is not a memory location dedicated to a particular processor but is merely an entry in a queue for a node, i.e. a group of processors. Thus, the processing unit of McKenney either spins on a node lock or a queue element. However, neither of these are a hand-off locking mechanism where a requesting processor spins on a memory location reserved for that processor until it is handed the lock by the owner.

Thus, McKenney does not teach each and every feature of independent claim 1, 11 and 21 as is required under 35 U.S.C. § 102(e). At least by virtue of their dependency on independent claims 1, 11 and 21, respectively, McKenney does not teach each and every feature of dependent claims 2-10, 12-20 and 22-30. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-30 under 35 U.S.C. § 102(e).

Furthermore, McKenney does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent the Examiner pointing out some teaching or incentive to implement McKenney such that, a hand-off lock is associated with the lock on the system resource if the attempt to obtain the lock is unsuccessful and the hand-off lock is obtained on the system resource if the attempt to obtain the lock on the system resource is unsuccessful, one of ordinary skill in the art would not be led to modify McKenney to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify McKenney in this manner, the presently claimed invention can be reached only through

an improper use of hindsight using the Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

Moreover, in addition to their dependency from independent claims 1, 11 and 21, respectively, McKenney does not teach the specific features recited in dependent claims 2-10, 12-20 and 22-30. For example, with regard to claims 3, 13 and 23, McKenney does not teach wherein the hand-off lock is a krlock. The Office Action states that McKenney teaches these features at column 12, lines 29-30. This section of McKenney teaches the kernel quad aware locking scheme (kqalock) described above, which is based on position in queue. The queue based locking scheme of McKenney has nothing to do with a krlock where a requesting processor spins on a memory location reserved for that processor until it is handed the krlock by the owner.

As a further example, with regard to claims 5, 15 and 25, the Office Action alleges that McKenney teaches the hand-off lock is obtained from a pool of hand-off locks at column 12, lines 32-33. This section of McKenney again teaches the kernel quad aware locking scheme (kqalock) described above, which is based on position in queue. There is nothing in this section, or any other section of McKenney, that teaches obtaining a hand-off lock, which is associated with a lock, from a pool of hand-off locks.

As an further example, with regard to claims 7, 17 and 27, the Office Action alleges that McKenney teaches obtaining the lock on the system resource, releasing the hand-off lock and handing-off the hand-off lock to a next processor spinning on the hand-off lock at column 12, lines 34-35 and 52-55. These sections again teach the kernel quad aware locking scheme (kqalock) which is based on position in queue. Once the processor is finished using the system resource, the processor whose request is next in queue then locks the resource. There is nothing in this section, or any other section of McKenney, that teaches releasing the hand-off lock, which is associated with the lock, and handing-off the hand-off lock to a next processor spinning on the hand-off lock.

As a further example, with regard to claims 8, 18 and 28, the Office Action alleges that McKenney teaches that a method is implemented in a multiprocessor system having one or more nodes, and wherein the hand-off lock includes a per-node word which contains a state of the hand-off lock on each node of the multiprocessor system and a per-processor spin field for each processor of the multiprocessor system at column

12, lines 28-29 and 52-53. In column 12, lines 28-29, McKenney teaches fairness control, which is the number of times the element has been used or an elapsed time, is used to check the queue element to see if it is available for conferring a preemptive position in the queue. In column 12, lines 52-53, McKenney teaches that when a lock release invoked with reference to a present queue element, the method decrements the spin and reference count of the queue element. There is nothing in this section, or any other section of McKenney, that teaches a method implementation in a multiprocessor system having one or more nodes, where the hand-off lock includes a per-node word which contains a state of the hand-off lock on each node of the multiprocessor system and a per-processor spin field for each processor of the multiprocessor system. As shown above, McKenney does not teach a hand-off lock.

As an additional example, with regard to claims 9, 19 and 29, the Office Action alleges that McKenney teaches when the lock on the system resource is released, the per-node word and per-processor spin fields of the hand-off lock are updated to reflect a next processor obtaining the lock on the system resource at column 12, lines 52-53. In this section McKenney teaches that when a lock release invoked with reference to a present queue element, the method decrements the spin and reference count of the queue element. There is nothing in this section, or any other section of McKenney, that teaches when the lock on the system resource is released, the per-node word and per-processor spin fields of the hand-off lock are updated to reflect a next processor obtaining the lock on the system resource. Again, McKenney does not teach a hand-off lock.

Therefore, in addition to being dependent on independent claims 1, 11 and 21 respectively, dependent claims 2-10, 12-20 and 3-30 are also distinguishable over McKenney by virtue of the specific feature recited in these claims. Accordingly, Applicants respectfully request withdrawal of the rejection of dependent claims 2-10, 12-20 and 3-30 under 35 U.S.C. § 102(e).

IV. Conclusion

It is respectfully urged that the subject application is patentable over McKenney and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: December 31, 2003

Francis Lammes

Francis Lammes
Reg. No. P-55,353
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Agent for Applicants

SJW/n